

## Lista zadań nr 13

### Zadanie 1. (3 pkt)

W serwisie SKOS znajduje się imperatywna wersja maszyny wirtualnej dla języka FUN. Rozszerz ją o implementację wyjątków. Instrukcje `BeginTry`, `EndTry` oraz `Raise` powinny działać w czasie stałym.

### Zadanie 2. (2 pkt)

Zmodyfikuj język MATCH tak, by konstruktory mogły przyjmować dowolnie wiele parametrów (w szczególności zero). Następnie wyraż unit oraz pary jako odpowiedni lukier składniowy.

### Zadanie 3. (3 pkt)

Rozszerz język MATCH o wzorce warunkowe, podobne do klauzul warunkowych w OCamlu. Wartość pasuje do wzorca  $p$  `when`  $e$  jeśli pasuje do wzorca  $p$ , a wyrażenie  $e$  oblicza się do prawdy. Wyrażenie  $e$  może korzystać ze zmiennych związanych przez  $p$ .

### Zadanie 4. (2 pkt)

Rozszerz język MATCH o alternatywę wzorców, podobną do tej z języka OCaml. Wartość pasuje do wzorca  $p_1 \mid p_2$  jeśli pasuje do  $p_1$  lub nie pasuje do  $p_1$  ale pasuje do  $p_2$ . Wzorce  $p_1$  i  $p_2$  powinny wiązać ten sam zestaw zmiennych.

### Zadanie 5. (2 pkt)

Rozszerz język MATCH o koniunkcję wzorców. Czy we wzorcu  $p_1 \& p_2$  podwzorce  $p_1$  i  $p_2$  powinny wiązać ten sam, czy rozłączny zestaw zmiennych? Czy negacja wzorców ma sens? A czy widzisz użyteczność tej konstrukcji?